

AMENDMENTS TO THE SPECIFICATION

Replace the Paragraph beginning at Page 6, line 14, with the following replacement paragraph:

The disclosed embodiment is directed to a fault-tolerant unified messaging system that enables initiation of messaging sessions for incoming callers, regardless of the incoming call type. In particular, a gateway such as a voice-over-IP gateway or a gateway serving as an interface between a telephony-based private branch exchange (PBX) and an application server, is configured for receiving an incoming call. The incoming call may be a voice call for a user attempting to call a messaging subscriber; alternatively the incoming call may be an incoming fax call from a fax machine for a subscriber that utilizes the called number for voice calls and fax messages. In response to the incoming call, the gateway sends requests to the appropriate messaging servers for initiation of messaging sessions for respective message types. The gateway simultaneously supports the multiple messaging sessions while determining the message type. Once the gateway determines the message type on the incoming call (e.g., DTMF digit indicating user-based call or fax tones indicating a fax call), the gateway sends a reject message to [[for]] the messaging sessions to be terminated due to their respective incompatible message types, and maintains the one messaging session having the matching message type. The application server, upon receiving the reject message, rolls back the necessary data structures such as event logs and events (i.e., deletes data from data structures) to the state prior to initiation of the messaging session, and terminates the instance executing the messaging session prior to execution of critical functions that are nonrecoverable.

Replace the Paragraph beginning at Page 10, line 18, with the following replacement paragraph:

Depending on the subscriber profile information, the instance may initiate a voice messaging session enabling a voice user to leave a message in an IMAP-based subscriber message store 32, or a ~~single-member~~ single-number reach session enabling a calling party to reach the subscriber via prescribed subscriber devices based on the application server 18 calling the prescribed subscriber devices attempting to reach the subscriber. The instance alternately may initiate an autoplay session to begin automatic playback of stored voice messages. Any different types of application sessions may be implemented in step 126, since the emphasis is to illustrate that the application server 18 has initiated the messaging operations based on the assumption that the incoming call is a voice call.

Replace the Paragraph beginning at Page 11, line 26, with the following replacement paragraph:

If in step 138 the application instance determines that the global variable specified that no fax was detected (Fax Detect = False), the application instance continues to commit and execute the critical operations in step 140. However if in step 138 the application instance determines that the global variable specified that a fax was detected by the gateway 12 (Fax Detect = True), the application instance begins rollback and termination operations. Specifically, the application instance deletes in step 142 all relevant data from the necessary data structures, for example any log entries, and any recorded messages sent by the gateway 12 for storage in the subscriber message store 32. The application instance also adds in step 144 a log entry specifying that operations were terminated due to a fax call, enabling management agents to trace the relevant logs of the application server to properly ~~identified~~ identify that the application session was properly terminated. The application instance then acknowledges in step 66 the H.225 Release Complete message, enabling tear down of the two-way RTP media stream 24, and terminates itself in step 146.

Replace the Paragraph beginning at Page 12, line 10, with the following replacement paragraph:

According to the disclosed embodiment, multiple application instances can be simultaneously initiated for concurrent execution of application operations for a single event, [[were]] where nonrelevant application instances can later be rolled back and terminated once the relevant application instances are identified. The disclosed arrangement is particularly effective for applications having antecedent steps that need to be performed before execution of critical operations, since additional information related to the critical operations can be obtained during the concurrent execution of the antecedent steps. For example, the disclosed arrangement may be applicable to billing verification processes that perform prescribed steps before execution of critical functions. The disclosed arrangement also may be effective for traffic management among multiple data paths, or even management of distributed processes across multiple platforms.